

Intelligent Water Distribution & Monitoring System

Prepared By: Kanchan Ganesh Dhuri

College: Vidylankar Institute of Technology

Category: Internet Of Things

Skills Required:

IBM IoT Platform, IBM Nodered, IBM Cloudant DB

Abstract:

The project Intelligent water distribution system, as the name says it is all about management of water supply throughout the scale, right from small societies, townships to entire urban infrastructure and also for irrigation water supply management. Main task of the water distribution system is to maintain the water in the tank and also generate the water bills to the individual households which involves human efforts. This system can be automated using the Internet of things.

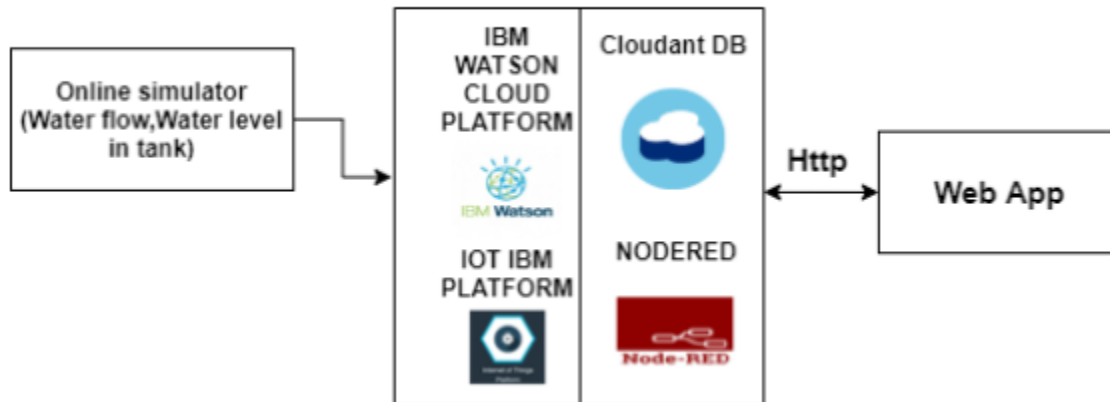
Solution Requirements:

The proposed system should continuously monitor the main tank water level and should automatically switch on/off the motors according to the tank water level and alert the admins. It should monitor the water flow of the individual houses and store the flow rate of each in the Cloudant DB to generate the water bills. Tank water level and the bills should be visualized in the dashboard so that the Admin can monitor them.

Project Flow:

- Main tank water level and Water flow to individual houses is continuously updated to IBM IoT platform (Use Online simulator sensor for water flow and water level)
- Create a Node-RED flow to get the data from IBM IoT platform and store it in cloudant DB.
- Display the tank water level in the UI
- Retrieve the flowrate of individual houses and generate bills and display them in UI.

Project Architecture:



Implementation:

Step1: IBM Academic Initiative Account is created

Step2: IBM watson IOT platform is created

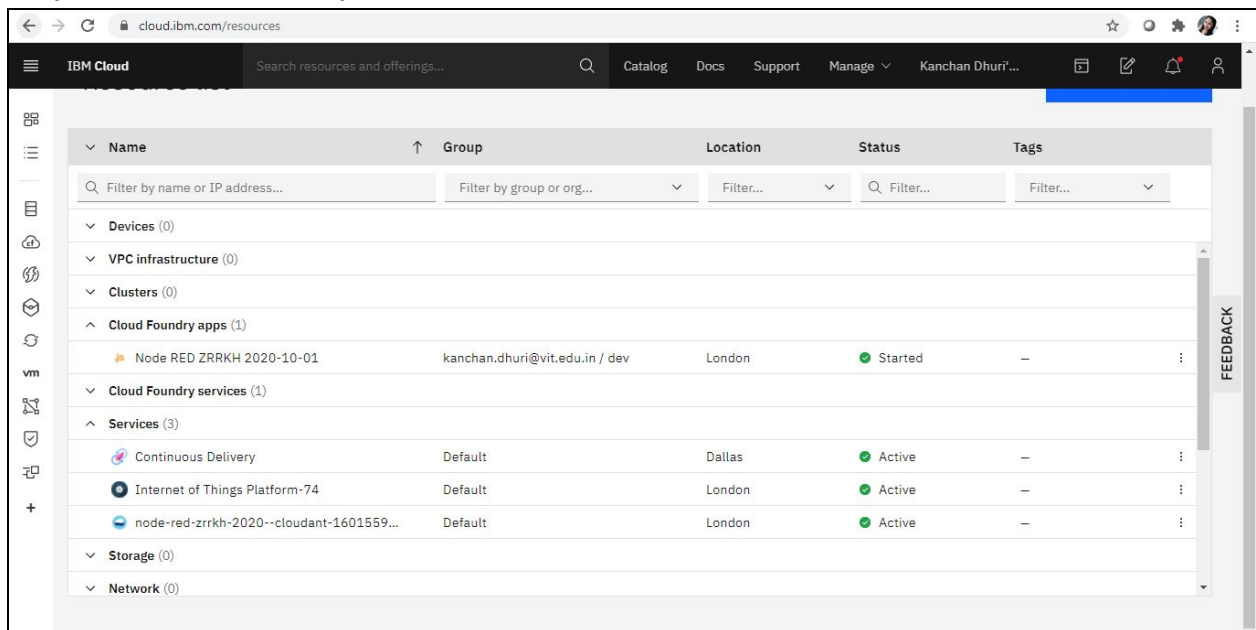
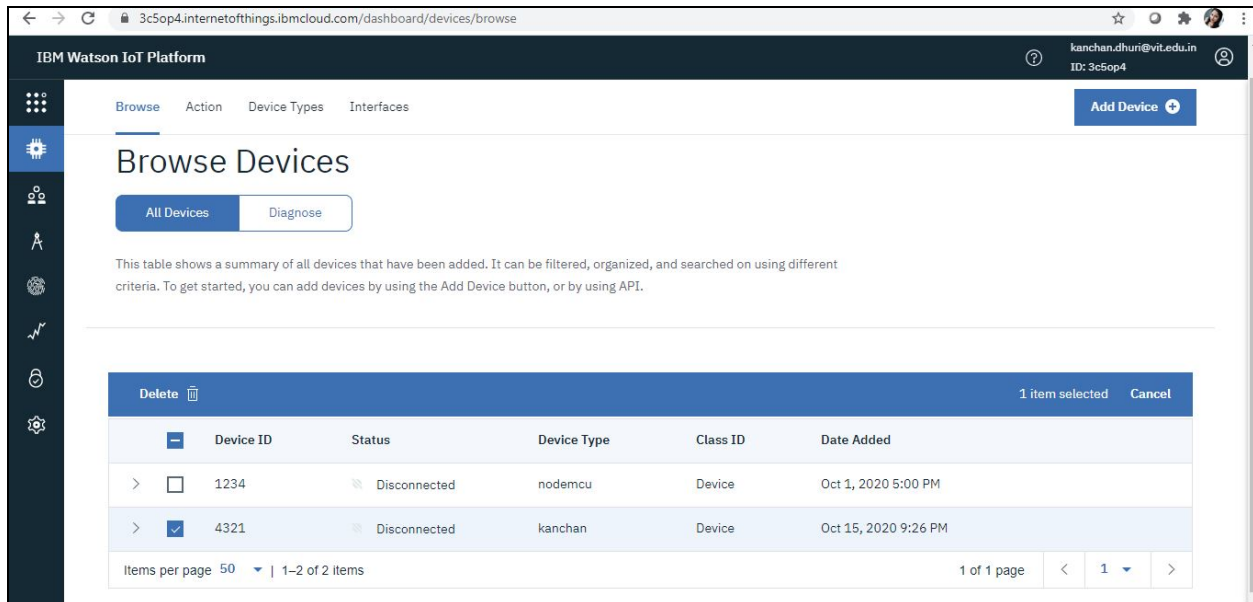


Fig1: on IBM cloud services are created : Internet of Things Platform-74 & Node-red

Step3: IOT platform configuration

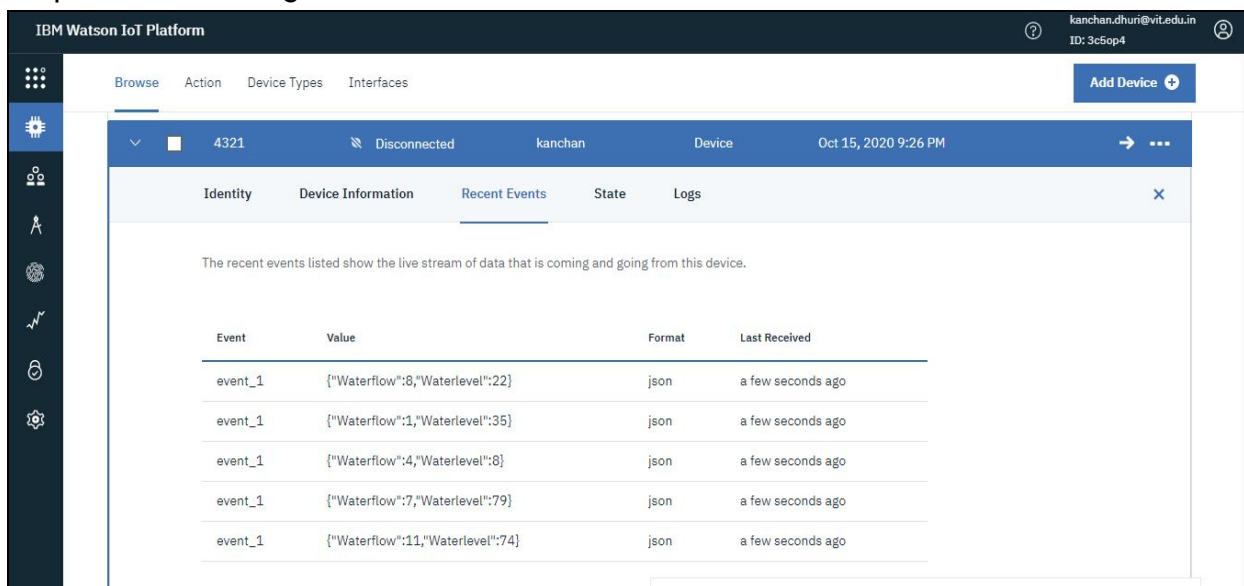


The screenshot shows the 'Browse Devices' page in the IBM Watson IoT Platform. The page has a sidebar with navigation icons and a top navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A 'Add Device' button is in the top right. Below the navigation, there's a 'Browse Devices' heading and two buttons: 'All Devices' (selected) and 'Diagnose'. A text block explains that the table shows a summary of all devices and can be filtered, organized, and searched. Below this is a table with columns: Device ID, Status, Device Type, Class ID, and Date Added. The table has two rows: one for device ID 1234 (Disconnected, nodemcu) and one for device ID 4321 (Disconnected, kanchan). The device ID 4321 is selected. At the bottom, there's a pagination bar showing '1 of 1 page' and '1-2 of 2 items'.

Device ID	Status	Device Type	Class ID	Date Added
1234	Disconnected	nodemcu	Device	Oct 1, 2020 5:00 PM
4321	Disconnected	kanchan	Device	Oct 15, 2020 9:26 PM

Fig2: IOT device is created and configured

Step4: Sensor configuration



The screenshot shows the 'Recent Events' page for device 4321 in the IBM Watson IoT Platform. The page has a sidebar with navigation icons and a top navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A 'Add Device' button is in the top right. Below the navigation, there's a 'Recent Events' heading and a table with columns: Event, Value, Format, and Last Received. The table has five rows of events, all with the format 'json' and 'a few seconds ago'.

Event	Value	Format	Last Received
event_1	{"Waterflow":8,"Waterlevel":22}	json	a few seconds ago
event_1	{"Waterflow":1,"Waterlevel":35}	json	a few seconds ago
event_1	{"Waterflow":4,"Waterlevel":8}	json	a few seconds ago
event_1	{"Waterflow":7,"Waterlevel":79}	json	a few seconds ago
event_1	{"Waterflow":11,"Waterlevel":74}	json	a few seconds ago

Fig3: Configured & connected online simulator sensors

The screenshot shows the 'Sensor configuration' page for a device named 'kanchan'. The page has a header with a back arrow, the device name 'Device Type: kanchan', and a user profile icon. Below the header, there's a section titled 'Events' with a count of '1' and a 'New event type +' button. The main configuration area includes:

- Event type name:** A text input field containing 'event_1', with a 'Send' button and a trash icon to its right.
- Schedule:** A section with a numeric input '30' and a dropdown menu set to 'Every Minute'.
- Payload:** A section with the instruction 'Specify the event payload in the editor window or by uploading a CSV file.' Below this is a code editor showing a JSON payload:


```
0 {
1   "Waterflow": random(0, 15),
2   "Waterlevel": random(0, 100)
3 }
4
```
- Buttons:** At the bottom left is 'Upload a CSV file'. At the bottom right are 'Cancel' and 'Save' buttons.

Fig4: Sensor configuration

Step5: Create & launch Node-Red Instance

The screenshot displays the IBM Cloud console interface for a specific resource. The top navigation bar includes the 'IBM Cloud' logo, a search bar, and links to 'Catalog', 'Docs', 'Support', and 'Manage'. The main content area shows the details for a resource named 'Node RED ZRRKH 2020-10-01', which is in a 'Running' state (indicated by a green dot). A sidebar on the left lists navigation options: 'Getting started', 'Overview' (selected), 'Runtime', 'Connections', 'Logs', 'API Management', and 'Autoscaling'. The main panel contains several widgets:

- Instances:** Shows 'Health' at '100%' with the note '1/1 instance(s) are running'. It also displays 'Instances' as '1' with a dropdown arrow.
- MB memory per instance:** A slider control ranging from '0' to '4352', with a current value of '256'.
- Runtime:** A section titled 'SDK for Node.js™' featuring a donut chart showing '256 Total MB allocation'. Below the chart, it states '7.75 GB still available' and includes a legend for 'Free' (light gray) and 'Used' (dark gray).
- Runtime cost:** A section with the text 'Current and estimated cost excludes connected services.'
- Connections (1):** A section showing a single connection with the ID 'node-red-zrrkh-2020--cloudant-1601559571665-74760'.

Fig5: Node-Red instance created & launched

Step6:Crete a Node-red flow to get data from IBM IoT device

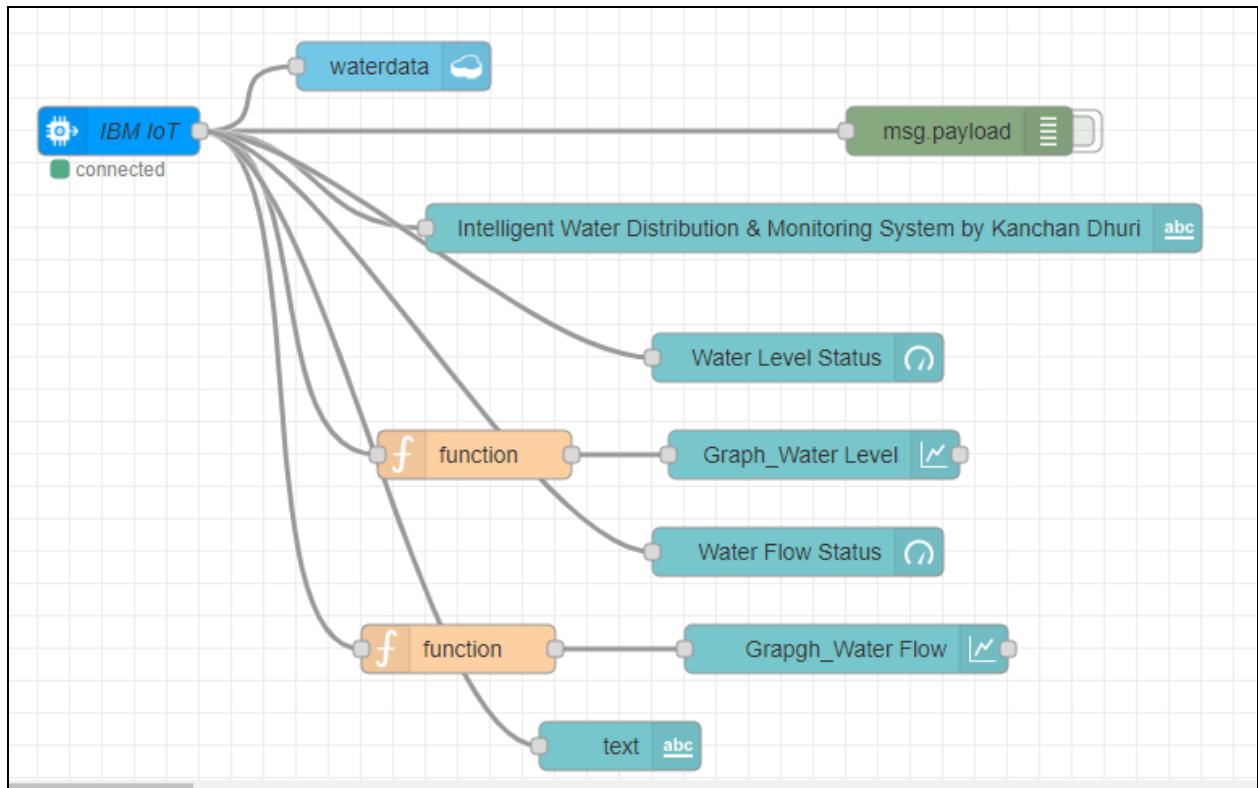


Fig6: Node-Red flow part1 to get & display sesnsor data on Web APP

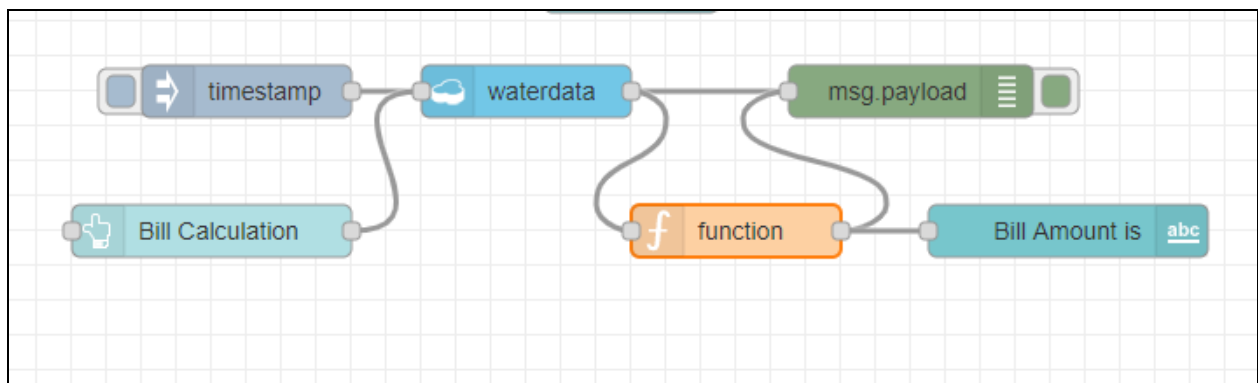


Fig7: Node-Red flow part2 to generate & display bill on Web APP

Step7: Configure the Node-Red flow to store data received from IBM IoT Device in Cloudant DB

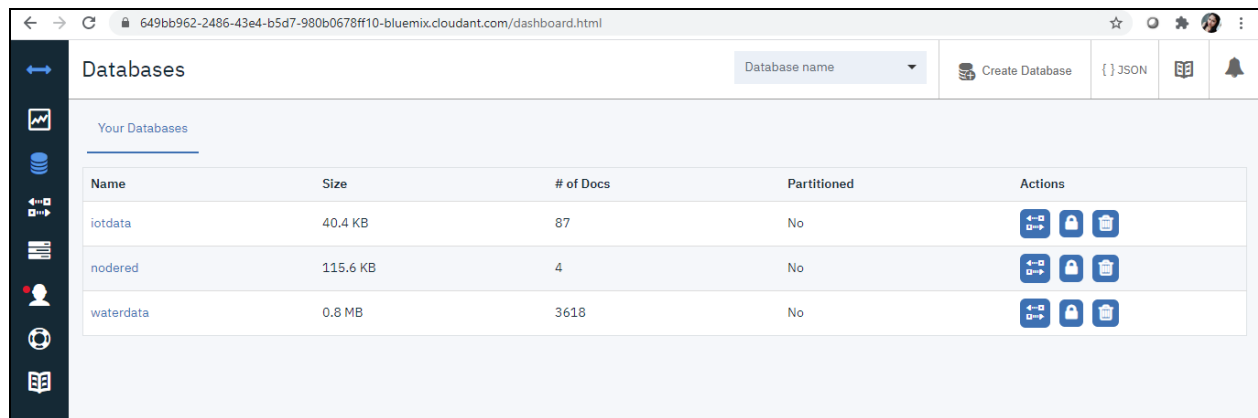


Fig8: Database is created named 'waterdata' to store sensor readings

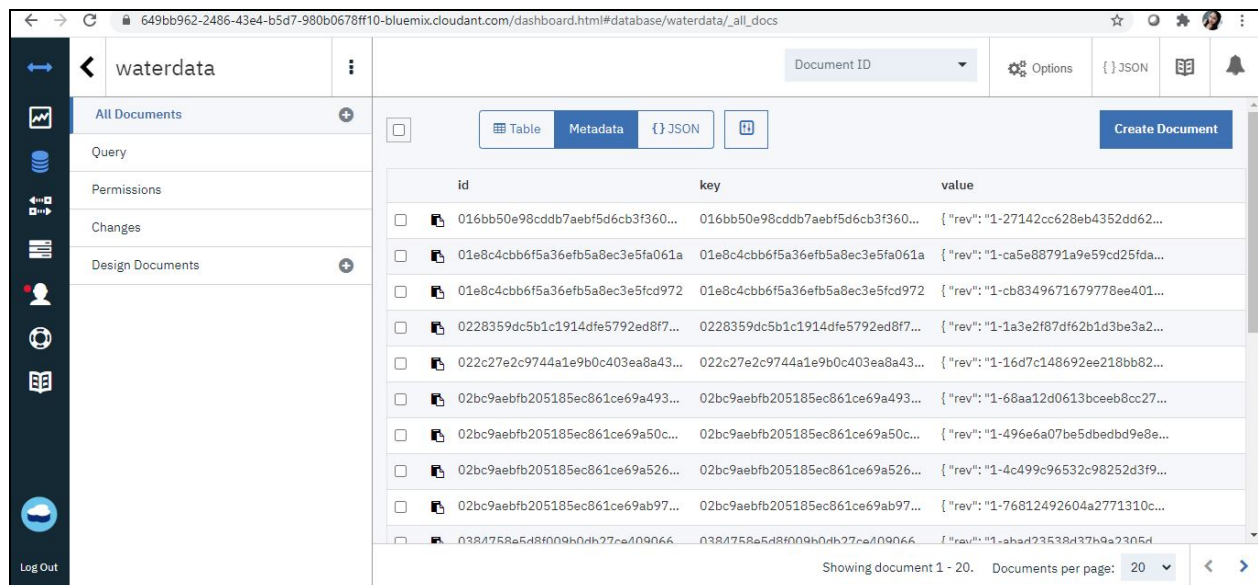


Fig9: Database files

Step8: Retrive monthly flowrate to calculate bill

Code:

```
//This is the function to multiply Waterflow parameter by 3 and
```

```
//add it to use one variable
```

```
valuelen=msg.payload.length
```

```
flow1=0;
```

```
for (i=0;i<valuelen;i++)
```

```
{
flow1=msg.payload[i].Waterflow+flow1;
}
```

```
bill= (flow1*3)/100;
```

```
msg.payload=bill;
return msg;
```

Step9: Use Dashboard nodes for creating Web App & Displaying Water flow rate, Water level & bill calculation

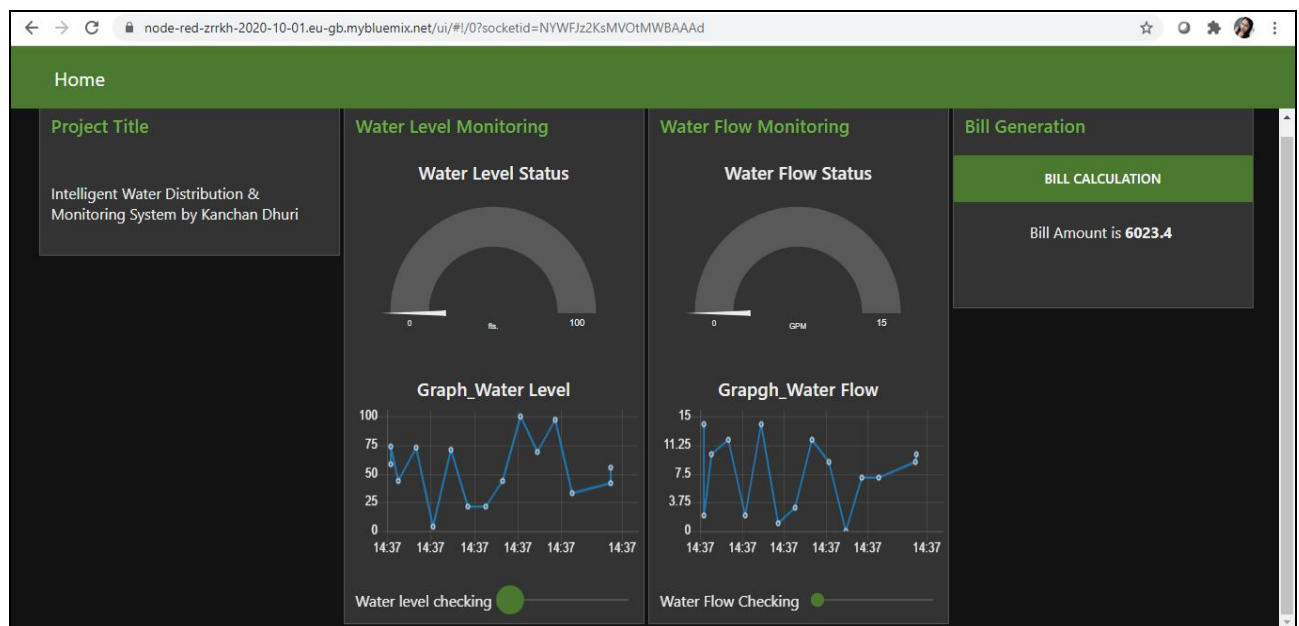


Fig10: Web APP

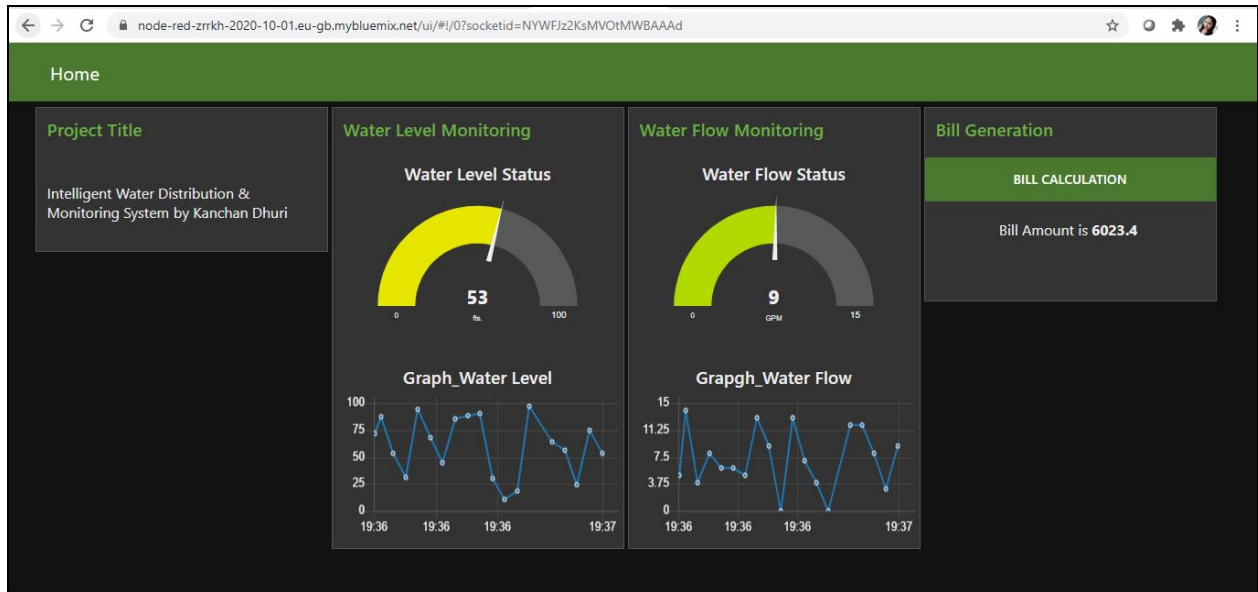


Fig11: Sensor readings & Bill Geneartion

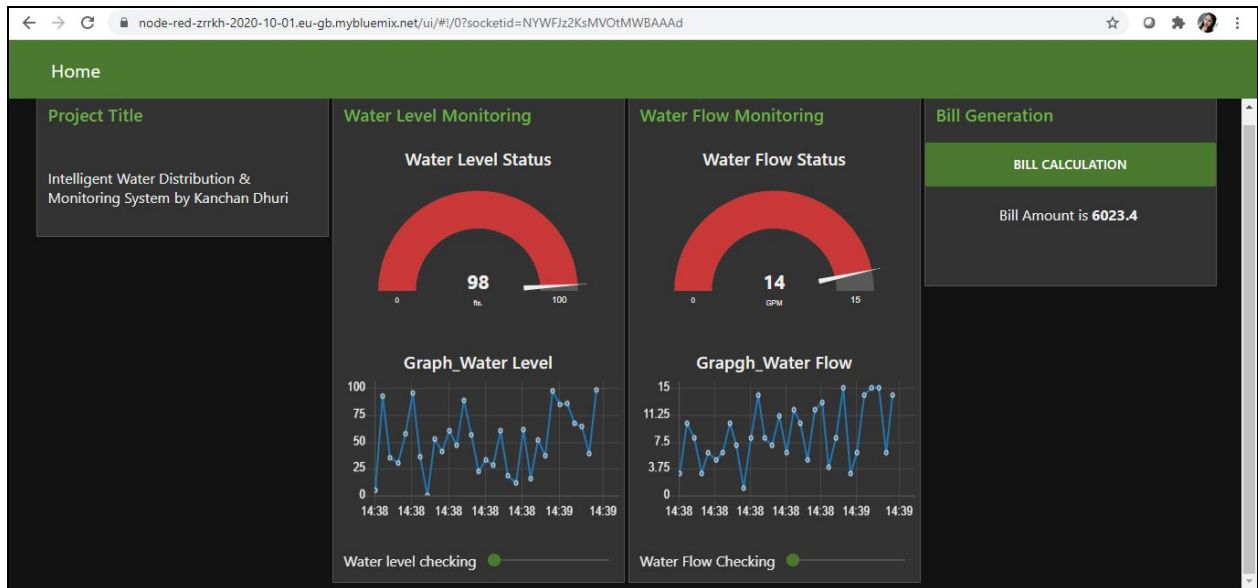


Fig12: Red color indicated Water level is above 90% of Total level of Tank

Step10: Results & Discussion

Here while monitoring water level different color code is used to indicated level of water, wether it is less than minimum threshold or greater than above threshold.

Red color indicated level more than 90fts and yellow color indicate level between 30to 80fts and green color indicate less than 30fts of the total level of Water Tank.

Bill is generated depending on water usage and water flow rate and considering cost 3/- per litre.

References:

[1] IBM Learning Resources